

WHAT IS CLAIMED IS:

- 1 1. An input/output hub, comprising:
- 2 an inbound ordering queue (IOQ) to receive inbound transactions, wherein all
- 3 read and write transactions have a transaction completion, peer-to-peer transactions are
- 4 not permitted to reach a destination until after all prior writes in the IOQ have been
- 5 completed, and a write in a peer-to-peer transaction does not permit subsequent accesses
- 6 to proceed until the write is guaranteed to be in an ordered domain of the destination;
- 7 an IOQ read bypass buffer to receive read transactions pushed from the IOQ to
- 8 permit posted writes and read/write completions to progress through the IOQ;
- 9 an outbound ordering queue (OOQ) to store outbound transactions and
- 10 completions of the inbound transactions, and to issue a write completion for a posted
- 11 write;
- 12 an OOQ read bypass buffer to receive read transactions pushed from the OOQ
- 13 to permit the posted writes and the read/write completions to progress through the OOQ;
- 14 and an unordered domain to receive the inbound transactions transmitted from the
- 15 IOQ and to receive the outbound transactions transmitted from an unordered protocol.

- 1 2. The input/output hub according to claim 1, wherein the IOQ does not permit the
- 2 inbound read and write transactions to bypass inbound write data.

- 1 3. The input/output hub according to claim 1, wherein the unordered protocol is a
- 2 coherence interface.

1 4. The input/output hub according to claim 3, wherein the coherent interface is a
2 Scalability Port.

1 5. An input/output hub, comprising:

2 an ordered domain, including:

3 an inbound ordering queue (IOQ) to receive and transmit inbound
4 transactions, wherein inbound read and write transactions are not permitted to
5 bypass inbound write data, all the read and write transactions have a transaction
6 completion, peer-to-peer transactions are not permitted to reach a destination until
7 after all prior writes in the IOQ have been completed, and a write in a peer-to-peer
8 transaction does not permit subsequent accesses to proceed until the write is
9 guaranteed to be in an ordered domain of the destination,

10 an IOQ read bypass buffer to receive read transactions pushed from the
11 IOQ to permit posted writes and read/write completions to progress through the
12 IOQ,

13 an outbound ordering queue (OOQ) to store outbound transactions and
14 completions of the inbound transactions, and to issue a write completion for a
15 posted write, and

16 an OOQ read bypass buffer to receive read transactions pushed from
17 the OOQ to permit the posted writes and the read/write completions to progress
18 through the OOQ; and

19 an unordered domain, in communication with an unordered protocol, including:

20 an inbound multiplexer to receive the inbound transactions from the
21 ordered domain to the unordered protocol, and
22 an outbound demultiplexer to receive the outbound transactions from the
23 unordered protocol to the ordered domain.

1 6. The input/output hub according to claim 5, further including at least one
2 Producer-Consumer ordered interface in communication with the ordered domain.

1 7. The input/output hub according to claim 6, further including an input/output
2 device connected with the Producer-Consumer ordered interface.

1 8. The input/output hub according to claim 7, further including an intermediary
2 device interconnecting the Producer-Consumer ordered interface and an input/output device.

1 9. The input/output hub according to claim 7, wherein the input/output device is a
2 Peripheral Component Interconnect (PCI) device.

1 10. The input/output hub according to claim 5, wherein the unordered protocol is a
2 coherence interface.

1 11. The input/output hub according to claim 10, wherein the coherent interface is a
2 Scalability Port.

12. An input/output system, comprising:

an ordered domain, including:

an inbound ordering queue (IOQ) to receive and transmit inbound transactions, wherein inbound read and write transactions are not permitted to bypass inbound write data, all the read and write transactions have a transaction completion, peer-to-peer transactions are not permitted to reach a destination until after all prior writes in the IOQ have been completed, and a write in a peer-to-peer transaction does not permit subsequent accesses to proceed until the write is guaranteed to be in an ordered domain of the destination;

an IOQ read bypass buffer to receive read transactions pushed from the IOQ to permit posted writes and read/write completions to progress through the IOQ;

an outbound ordering queue (OOQ) to store outbound transactions and completions of the inbound transactions, and to issue a write completion for a posted write;

an OOQ read bypass buffer to receive read transactions pushed from the OOQ to permit the posted writes and the read/write completions to progress through the OOQ;

an unordered domain, in communication with an unordered protocol, including:

an inbound multiplexer to receive the inbound transactions from the ordered domain to the unordered protocol, and

an outbound demultiplexer to receive the outbound transactions from the unordered protocol to the ordered domain;

24 a Producer-Consumer ordered interface in communication with the ordered
25 domain;
26 an input/output device connected with the Producer-Consumer ordered
27 interface; and
28 a coherent interface within the unordered protocol in communication with the
29 unordered domain.

1 13. The input/output system according to claim 12, wherein the coherent interface is a
2 Scalability Port.

1 14. The input/output system according to claim 12, wherein the input/output device is
2 a Peripheral Component Interconnect (PCI) device.

1 15. The input/output system according to claim 12, further including an intermediary
2 device interconnecting the Producer-Consumer ordered interface and the input/output device.

1 16. An input/output system, comprising:

2 an ordered domain having a first functional block and a second functional block,
3 wherein the first functional block and the second functional block each include:

4 an inbound ordering queue (IOQ) to receive inbound transactions, wherein
5 inbound read and write transactions are not permitted to bypass inbound write
6 data, all the read and write transactions have a transaction completion, peer-to-
7 peer transactions are not permitted to reach a destination until after all prior writes

8 in the IOQ have been completed, and a write in a peer-to-peer transaction does
9 not permit subsequent accesses to proceed until the write is guaranteed to be in an
10 ordered domain of the destination,

11 an IOQ read bypass buffer to receive read transactions pushed from the
12 IOQ to permit posted writes and read/write completions to progress through the
13 IOQ,

14 an outbound ordering queue (OOQ) to store outbound transactions and
15 completions of the inbound transactions, and to issue a write completion for a
16 posted write,

17 an OOQ read bypass buffer to receive read transactions pushed from the
18 OOQ to permit the posted writes and the read/write completions to progress
19 through the OOQ;

20 an unordered domain, in communication with an unordered protocol, including:

21 an inbound multiplexer to receive the inbound transactions from the
22 ordered domain to the unordered protocol, and

23 an outbound demultiplexer to receive the outbound transactions from the
24 unordered protocol to the ordered domain;

25 a first Producer-Consumer ordered interface in communication with the first
26 functional block;

27 a first input/output device connected with the first Producer-Consumer ordered
28 interface;

29 a second Producer-Consumer ordered interface in communication with the second
30 functional block;

31 a second input/output device connected with the second Producer-Consumer
32 ordered interface; and
33 a coherent interface within the unordered protocol in communication with the
34 unordered domain.

1 17. The input/output system according to claim 16, wherein the coherent interface is a
2 Scalability Port

1 18. The input/output system according to claim 16, wherein the first input/output
2 device is a Peripheral Component Interconnect (PCI) device.

1 19. The input/output system according to claim 16, wherein the second input/output
2 device is a Peripheral Component Interconnect (PCI) device.

1 20. The input/output system according to claim 16, further including a first
2 intermediary device interconnecting the first Producer-Consumer ordered interface and the first
3 input/output device.

1 21. The input/output system according to claim 16, further including a second
2 intermediary device interconnecting the second Producer-Consumer ordered interface and the
3 second input/output device.

1 22. A computer system, comprising:

2 a plurality of processor units having access to caches;
3 a main memory;
4 a coherent interface to maintain coherency between the processor units and their
5 caches;

6 a scalability node controller interconnecting the processor units, the main
7 memory, and the coherent interface to control interface therebetween; and

8 an input/output hub in communication with the coherent interface, including:

9 an inbound ordering queue (IOQ) to receive inbound transactions, wherein
10 all read and write transactions have a transaction completion, peer-to-peer
11 transactions are not permitted to reach a destination until after all prior writes in
12 the IOQ have been completed, and a write in a peer-to-peer transaction does not
13 permit subsequent accesses to proceed until the write is guaranteed to be in an
14 ordered domain of the destination;

15 an IOQ read bypass buffer to receive read transactions pushed from the
16 IOQ to permit posted writes and read/write completions to progress through the
17 IOQ;

18 an outbound ordering queue (OOQ) to store outbound transactions and
19 completions of the inbound transactions, and to issue a write completion for a
20 posted write;

21 an OOQ read bypass buffer to receive read transactions pushed from the
22 OOQ to permit the posted writes and the read/write completions to progress
23 through the OOQ; and

25

2

2